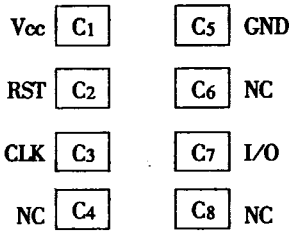
逻辑加密存储 SLE4442 卡及其应用

1、概述

SLE4442 是德国西门子（SIMENS）公司设计的逻辑加密存储卡。它具有 2K 位的存储容量和完全独立的可编程代码存储器（PSC）。内部电压提升电路保证了芯片能够以单+5V 电压供电，较大的存储容量能够满足通常应用领域的各种需要。因此是目前国内应用较多的一种 IC 卡芯片。芯片采用多存储器结构，2 线连接协议（串行接口满足 ISO7816 同步传送协议），NMOS 工艺技术，每字节的擦除/写入编程时间为 2.5ms。存储器具有至少 10000 次的擦写周期，数据保持时间至少 10 年。

芯片引脚

SLE4442 的触点安排见下图



起引脚的定义和功能说明如下表

引脚	卡触点	符号	功能
1	C1	V_{cc}	操作电压 5V
2	C2	RST	复位
3	C3	CLK	时钟
4	C4	NC	未用
5	C5	GND	地
6	C6	NC	未用
7	C7	I/O	双向数据线(漏极开路)
8	C8	NC	未用

芯片功能

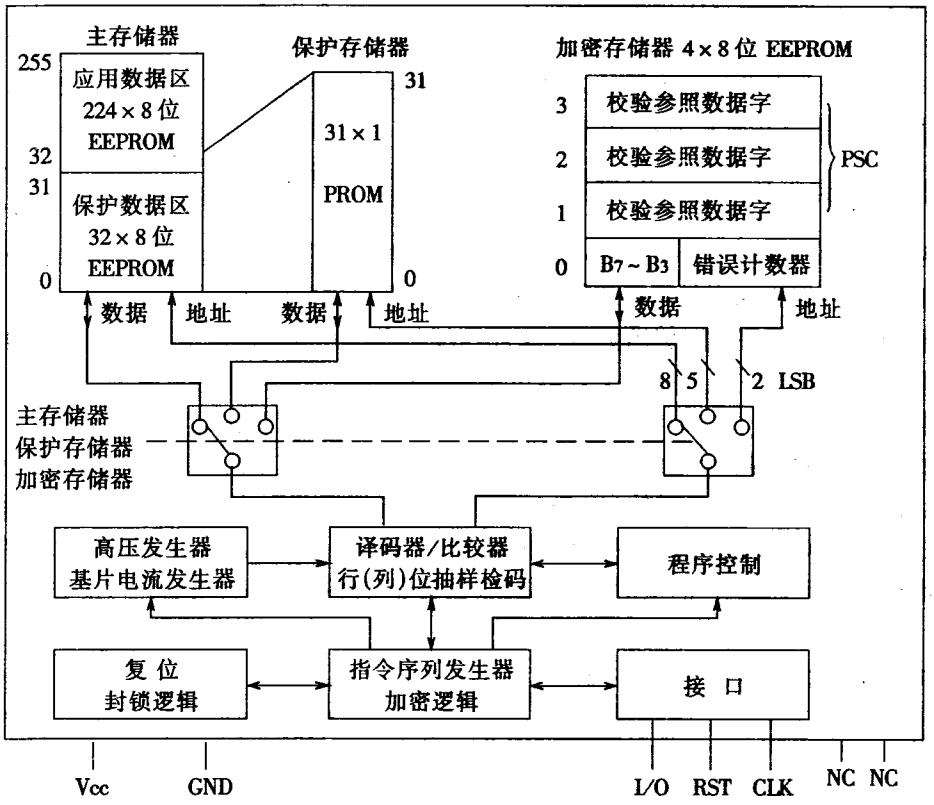
SLE4442 IC 卡主要包括三个存储器 ：

（1）256x8 位 EEPROM 型主存储器。地址 0~31 为保护数据区，该区数据读出不受限制，写入受保护存储内部数据状态的限制。当保护存储器中第 N 位（N=0~31）为 1 时，对应主存储器中第 N 个字节允许进行擦除和写入操作。地址 32~255 后 244 字节为应用数据区 ，数据读出不受限制，擦除和写入受加密存储器数据校验结果的影响。这种 加密校验的控制是对整个主存储器实施的（即包括保护数据区和应用数据区）。

（2）32 x1 位 PROM 型保护存储器。一次性编程以保护主存储器保护数据区，防止一些固定的标识参数被改动。保护存储器同样受加密存储器数据校验结果的影响。

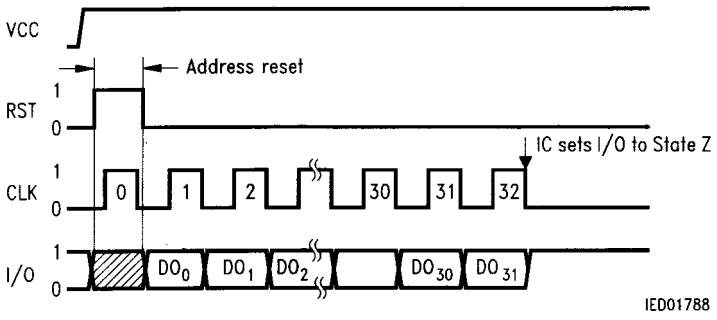
（3）4x8 位 EEPROM 型加密存储器。第 0 字节为密码输入错误计数器（EC）。EC 的有效位 是低三位，芯片初始化时设置成“111”。这一字节是可读的。EC 的 1，2，3 字节为参照字存储区。这 3 个字节的内容作为一个整体被称为可编程加密代码（PSC）。其读出，写入和擦除均受自身“比较”操作结果的控制。

芯片内部逻辑结构



传送协议

（1）复位和复位响应：



复位和复位响应是根据 ISO7816-3 标准来进行的。在操作期间的任意时候都可以复位。开始，地址计数器随一个时钟脉冲而被设置为零。当 RST 线从高状态（H）置到低状态（L）时,第一个数据位(LSB)的内容被送到 I/O 上。若连续输入 32 个时钟脉冲，主存储器中的前四个字节地质单元中的内容被读出。在第 33 个始终脉冲的下降沿，I/O 线被置成高状态而关闭。

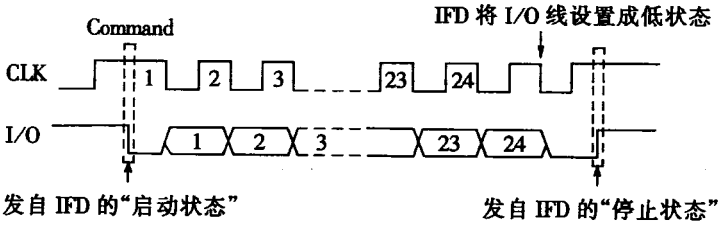
（2）命令模式：

复位响应以后，芯片等待着命令。每条命令都以一个“启动状态”开始。整个命令包括 3 个字节。随后经更着一个附加脉冲并用一个“停止状态”来结束操作。

启动状态：在 CLK 为高状态（H 状态）期间，I/O 显得下降沿为启动状态。

停止状态：在 CLK 为高状态（H 状态）期间，I/O 显得上升沿为停止状态。

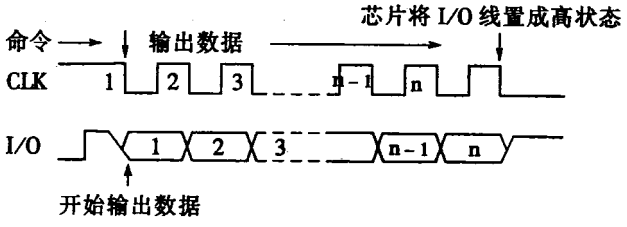
在接受一个命令之后，有两种可能的模式：输出数据模式（即读数据）和处理数据模式。



命令模式的时序关系

（3） 输出数据模式:

这种模式是将 IC 卡芯片中的数据传送个外部设备接口(IFD)的一种操作。



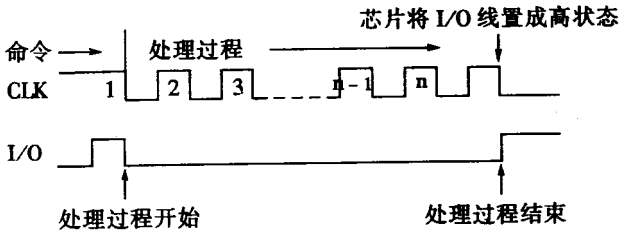
输出数据模式的时序关系

在第一个 CLK 脉冲的下降沿之后，I/O 线上的第一位数据变为有效。随后每增加一个时钟脉冲，芯片内部的一位数据被送到 I/O 线上。 骑术除数叙事从每个字节的最低位（LSB）开始。当说需要的最后一个数据送出以后，需要在附加一个时钟脉冲来把 I/O 置成高状态，以便接受新的命令。

在输出数据期间，任何“启动状态”和“停止状态”均被屏蔽掉。

（4）处理数据模式：

这种模式是对 IC 芯片作内部处理。



处理模式的时序关系

芯片在第一个始终脉冲的下降沿，将 I/O 线从高状态拉到低状态并开始处理。此后芯片在内部连续计时计数，直到低 n 个始终脉冲之后的附加一个时钟脉冲的下降沿 I/O 线再次置高，完成芯片的处理过程。在整个处理过程中 I/O 线被锁定成低状态。

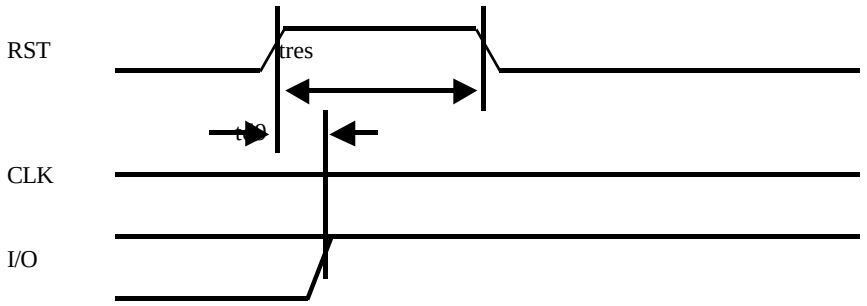
芯片的复位方式

（1） 外部复位：SLE4442 时基于同步夫位响应的传送协议。芯片的副位时序如前述。

（2） 加电复位：在把操作电压连接到 Vcc 段之后，芯片内部进行复位操作。I/O 线被置为高状态。必须在

对任意地址进行读操作或做一个复位响应操作之后才可以进行数据交换。

（3）中止：在 CLK 为低状态期间，如果 RST 置为高状态，则任何操作均无效。I/O 线被锁定到高状态。需要一个最小维持时间 tres=5Us 之后，芯片才能接受新的有小复位，中止状态的时序关系如下图。中止状态之后，芯片又准备下一个操作。



芯片的操作命令

（1） 命令格式：

每条命令包含三个字节，其排列顺序如下：

MSB 控制字 LSB	MSB 地址字 LSB	MSB 数据字 LSB
B7 B6 B5 B4 B3 B2 B1 B0	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0

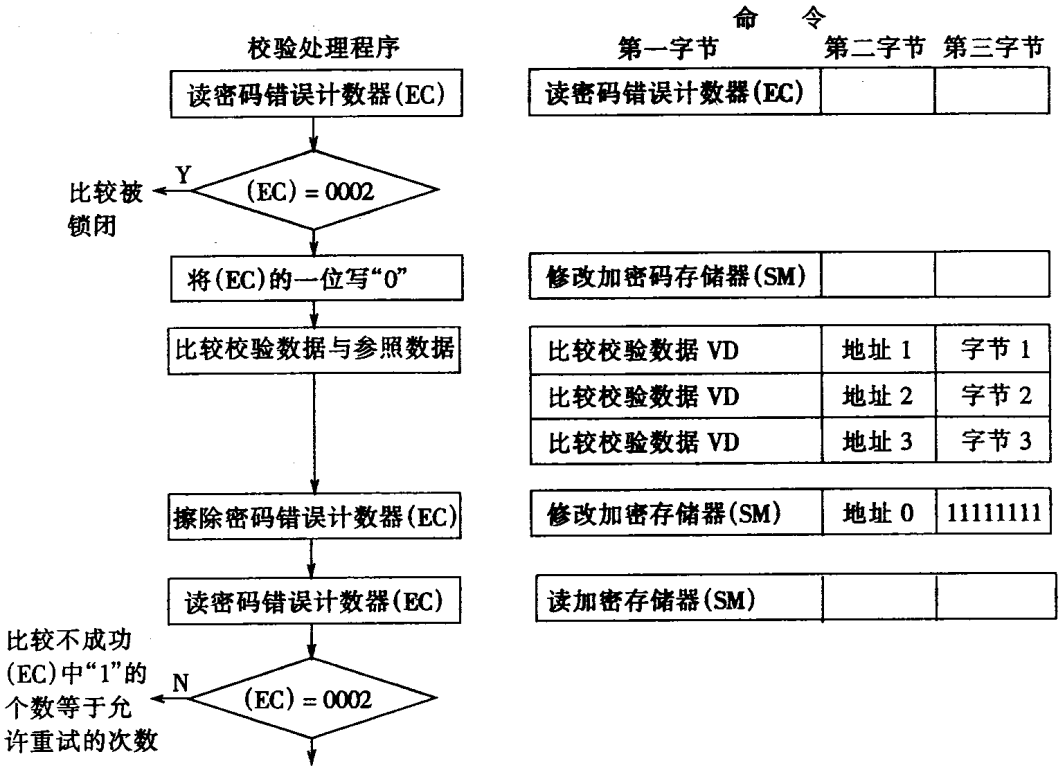
（2） SLE4442 芯片具有七种命令，其格式和功能见表。

字节 1（控制） B7~B0	字节 2（地址） A7~A0	字节 3（数据） D7~D0	功能	命令模式
30H	地址数	无效	读主储存器	输出数据模式
38H	地址数	输入数据	修改主储存器	处理模式
34H	无效	无效	读保护储存器	输出数据模式
3CH	地址数	输入数据	写保护储存器	处理模式
31H	无效	无效	读加密储存器	输出数据模式
39H	地址数	输入数据	修改加密储存器	处理模式
33H	地址数	输入数据	比较校验数据	处理模式

注意：对于每个字节来说总是从最地位 LSB 开始读出。写入时首先传送的也是字节的最低为（LSB）。

对保护存储器进行修改时，输入数据必须与原有数据相等，才能正确保护。

比较校验数据流程如下。



比较校验数据的程序流程

具体操作参照程序

```
;*****************************************************************************
                    SLE4442  通用读写模块使用说明
```

ReadCard 读出从 StartAdr 开始的 ByteNum 字节,结果存放于 ReadBuf 中,读出正确, ACC 中返回#0,返回#01 表示无效卡或者卡损坏。

WriteCard 从 StartAdr 开始写入 ByteNum 字节,待写入数据存放于 WriteBuf 中,写入正确 ACC 中返回#0,返回#01 表示无效卡或者卡损坏。

写卡之前一定要调用 CheckPassword 核对密码,密码存放于 PSW 开始的三个单元,ACC 中返回#0 表示核对正确,#01 表示无效卡或者卡损坏,#02 表示密码错,#03H 表示卡已被锁闭,#04H 表示该卡还有一次试验机会。核对密码完成后,EC 开始的 4 个单元返回错误计数值及密码。

若要对保护存储区进行写保护,请调用 WriteProtect 子程序,参数及返回值与 WriteCard 相同。

读保护存储器调用 ReadProtect,参数及返回值同 ReadCard。

```
;*****************************************************************************
;编制:尹寒冬
;***********************************
;Variabal
        ReadBuf   EQU    30H
        WriteBuf   EQU    40H
        EC        EQU    50H          ;4 字节临时单元
```

```
          PSWD      EQU     54H          ;3 字节密码单元
          ByteNum   EQU     57H
          StartAdr     EQU     58H
;Bit Variabal
          RST       BIT     P0.7
          IO          BIT     P1.7
          CLK       BIT     P1.6
          PWR       BIT     P0.6         ;IC 卡上电位,子程序中未用
;Constant
          Idetify1      EQU     0A2H          ;此处为保护区 00~03 单元的值,用于识别卡
          Idetify2      EQU     13H
          Idetify3      EQU     10H
          Idetify4      EQU     91H
;*****************************************
;从主存储器中读出数据块
ReadCard: LCALL ANRST
         JNZ    ReadExit

         MOV    R0,#ReadBuf
         MOV    R2,ByteNum
         MOV    R5,StartAdr
         LCALL   Remm
         MOV    A,#00H
ReadExit: RET
;*****************************************
;修改主存储器中的数据块
WriteCard:
         LCALL   ANRST
         JNZ      WriteExit
         MOV     R0,#WriteBuf
         MOV     R2,ByteNum
         MOV     R5,StartAdr
WriteLoop:
         MOV     A,@R0
         MOV     R6,A
         LCALL    Wrmm
         INC       R0
         INC       R5
         DJNZ      R2,WriteLoop
         MOV      A,#00H
WriteExit:
         RET
;*****************************************
;从保护存储器中读出数据块（32 位），R2 最大为 4
```

```
ReadProtect:
        LCALL ANRST
        JNZ   RpExit

        MOV   R0,#ReadBuf
        MOV   R2,ByteNum
        MOV   R5,StartAdr
        LCALL  Repm
        MOV   A,#00H
RpExit:
        RET
;*******************************************
;保护保护数据区（32 字节）中的数据块，参考数据必须与原数据相等
WriteProtect:
        LCALL  ANRST
        JNZ     WpExit
        MOV    R0,#WriteBuf
        MOV    R2,ByteNum
        MOV    R5,StartAdr
WpLoop:
        MOV    A,@R0
        MOV    R6,A
        LCALL   Wrpm
        INC      R0
        INC      R5
        DJNZ     R2,WpLoop
        MOV     A,#00H
WpExit:
        RET

;*********************************
;该程序只允许两次重试，以避免卡损坏。
CheckPassword:
         LCALL   ANRST
         JNZ      CheckExit
         LCALL   Rescm              ;读加密存储器，获取错误计数器值
         MOV     A,EC
         ANL      A,#07h
         JZ        IsBadCard          ;EC=0，卡已报废
         RRC      A
         JNC      ChkPsw1
         MOV     A,EC
         ANL      A,#06h             ;将 EC 的一位写 “0”
         JZ        HaveOneChance      ;该卡还有一次试验机会
```

```
        SJMP    ChkPsw2
ChkPsw1:
        RRC     A
        JNC     HaveOneChance       ;该卡还有一次试验机会
        MOV     A,EC
        ANL     A,#05H
        JZ      HaveOneChance       ;该卡还有一次试验机会
ChkPsw2:
        MOV     R5,#00H
        MOV     R6,A
        LCALL   Wrscm               ;将 EC 写回卡
        MOV     R2,#3
        MOV     R5,#01H             ;卡内密码首址
        MOV     R0,#PSWD            ;参考数据首址
ChkPswLoop:
        MOV     A,@R0
        MOV     R6,A
        LCALL   Verda               ;开始核对密码
        INC     R0
        INC     R5
        DJNZ    R2, ChkPswLoop
        MOV     R5,#00H
        MOV     R6,#0FFH
        LCALL   Wrscm               ;擦除错误计数器
        LCALL   Rescm               ;读错误计数器以检查核对是否成功
        MOV     A,EC
        ANL     A,#07H
        CJNE    A,#07H,IS_FAIL      ;错误计数器不能擦除，核对密码失败
        MOV     A,#00H              ;错误计数器正确擦除，核对密码成功
CheckExit:
        RET
IS_FAIL:                            ;核对密码失败
        MOV   A,#02H
        RET
IsBadCard:
        MOV   A,#03H
        RET
HaveOneChance：
        MOV   A,#04H
        RET

;*************************************************************************
;R4,R5,R6 分别是命令、地址、数据字节
;*********************************
```

```
SendComm:
          SETB    IO                    ;产生开始状态
          LCALL   Delay10uS
          SETB    CLK
          LCALL   Delay5uS
          CLR      IO
          LCALL   Delay5uS
          CLR     CLK
                                        ;发送命令
          MOV     A,R4
          LCALL    SendByte
          MOV      A,R5
          LCALL    SendByte
          MOV      A,R6
          LCALL    SendByte
                                        ;产生停止状态
          CLR       IO
          LCALL     Delay5uS
          SETB      CLK
          LCALL     Delay5uS
          SETB       IO
          LCALL     Delay5uS
          RET
;****************************************
SendByte:                                ;发送一个字节数据
          MOV       R3,#8
SendLoop:
          RRC       A
          MOV       IO,C
          LCALL     Delay10uS           ;
          SETB      CLK
          LCALL     Delay10uS
          CLR       CLK
          DJNZ      R3,SendLoop
          RET
;************************************************
;R2--Byte number, R5--Start adress,@r0--return data
;************************************************
Repm:                                   ;设置读保护存储器命令
          MOV      R4,#34H
          SJMP      RmStart
Remm:                                   ;设置读主存储器命令
          MOV      R4,#30H
RmStart:
```

```
          LCALL     SendComm
RmLoop:
          CLR       A
          MOV       R3,#8
RmByte:
          CLR       CLK
          NOP
          NOP
          NOP
          NOP
          SETB      IO
          NOP
          MOV      C,IO
          RRC       A
          NOP
          NOP
          SETB      CLK
          LCALL     Delay10uS
          DJNZ       R3, RmByte
          MOV       @R0,A
          INC         R0
          DJNZ        R2, RmLoop
          LCALL      Break               ;中止操作
          RET

;*********************************************************************
Break:     CLR         CLK
          LCALL      Delay5uS
          SETB        RST
          LCALL       Delay5uS
          CLR         RST
          RET
;*********************************************************************
Verda:
          MOV         R4,#33H            ;设置比较校验数据命令
          SJMP        WrStart
Wrscm:
          MOV         R4,#39H            ;设置修改加密存储器命令
          SJMP        WrStart
Wrpm:
          MOV         R4,#3CH            ;设置修改保护存储器命令
          SJMP          WrStart
Wrmm:
          MOV          R4,#38H            ;设置修改主存储器命令
```

```
WrStart:
          LCALL       SendComm
WrmOption:
          CLR         CLK
          NOP
          NOP
          NOP
          NOP
          SETB         IO
          NOP
          MOV        C,IO
          JC           WrmOK            ;如果 I/O 返回高状态，中止处理过程
          NOP
          NOP
          NOP
          NOP
          SETB        CLK
          LCALL       Delay10uS
          SJMP        WrmOption
WrmOK:
          RET
;*********************************************************************
Rescm:    MOV      R4,#31H              ;设置读加密存储器命令
          LCALL     SendComm
          MOV      R2,#4
          MOV      R0,#EC
rescm2:
          MOV       R3,#8
rescm1:    CLR       CLK
          NOP
          NOP
          NOP
          NOP
          SETB       IO
          NOP
          MOV       C,IO
          RRC        A
          NOP
          NOP
          SETB      CLK
          LCALL     Delay10uS
          DJNZ       R3,rescm1
          MOV       @R0,a
          INC        R0
```

```
          DJNZ      R2,rescm2
          CLR       CLK
          LCALL     Delay5uS
          RET

;*****************************************************************************
Anrst:
          MOV       R0,#EC
          SETB      RST                ;产生复位响应时序
          LCALL     Delay5uS            ;
          SETB      CLK
          LCALL     Delay10uS          ;
          CLR       CLK
          LCALL     Delay5uS           ;
          CLR       RST
                                        ;接受复位响应值
          MOV       R2,#4
anrst1:     MOV     R3,#8
anrst2:     LCALL    Delay10uS           ;
          SETB      CLK
          SETB      IO
          NOP
          NOP
          MOV       C,IO
          RRC       A
          LCALL     Delay5uS
          CLR       CLK
          DJNZ      R3,anrst2
          MOV       @R0,A
          INC       R0
          DJNZ      R2,anrst1

;以下代码根据复位响应返回值判断卡的合法性及有效性
          MOV   R0,#EC
          MOV   A,@R0
          CJNE   A,#Idetify1,CardErr
          INC     R0
          MOV   A,@R0
          CJNE   A,#Idetify2,CardErr
          INC     R0
          MOV   A,@R0
          CJNE   A,#Idetify3,CardErr
          INC     R0
          MOV   A,@R0
```

```
          CJNE   A,#Idetify4,CardErr
          MOV    A,#00H
          RET
CardErr:   MOV   A,#01H
          RET
;*****************************************
Delay10uS:                          ;延时 10Us
         MOV    R7,#06H
         DJNZ   R7,$
         RET
;*****************************************
Delay5uS:                           ;延时 5Us
         MOV    R7,#02H           ;
         DJNZ   R7,$
         RET
;*****************************************
```

2、汇编软件包

```
;SLE4442 操作软件包       (V0.1)
;软件包的接口指令有：读主存储器，写主存储器，读保护存储器，写
;保护存储器，读保护数据区，写保护数据区，校验密码子程序.

;定义 DELAY5uS 和 DELAY5uS
DELAY5uS          MACRO          宏定义
                  NOP            ;12MHZ 晶振，1 机器周期 1uS
                  NOP
                  NOP
                  NOP
                  NOP
                  ENDM
;卡的复位和复位响应程序
;出口数据是卡号，存在 CARDBAK 缓冲区(4 字节)
ANRST:
                  SETB       RST
                  DELAY5uS
                  SETB       CLK
                  DELAY5uS
```

```
                CLR     CLK
                DELAY5uS
                CLR     RST
                MOV     R2,#04H         读出四个字节
ANRST_L1:       MOV     R3,#08H
ANRST_L2:       DELAY5uS                复位响应，读出卡号
                SETB    CLK             即主存储器前四个字节地质单元。
                SETB    IO
                NOP
                NOP
                MOV     C,IO            数据在上升沿有效。
                RRC     A
                DELAY5uS
                CLR     CLK
                DJNZ    R3,ANRST_L2     读取一字节数据位
                DJNZ    R2,ANRST_L1
                RET

;发送字节数据子程序
;发送数据字节在 ACC
SENDBYTE:
                MOV     R3,#08H
SENDBYTE_L1:    RRC     A
                MOV     IO,C            先发送低位
                DELAY5uS
                SETB    CLK
                DELAY5uS
                CLR     CLK
                DJNZ    R3,SENDBYTE_L1
                RET

;读字节数据子程序
;发出读一个字节的脉冲，返回在 ACC
RCVBYTE:
                CLR     A
                MOV     R3,#08H
RCVB_L1:        CLR     CLK
                DELAY5uS
                SETB    IO
                NOP
                NOP
                MOV     C,IO
                RRC     A
                NOP
```

```
        NOP
        SETB    CLK
        DELAY5uS
        DJNZ    R3,RCVB_L1
        RET
```

;**中止操作子程序**

;**因为读出是连续的，固读主存储器时要调用此子程序中止读操作**

```
BREAK:
        CLR     CLK     产生中止时序
        DELAY5uS
        SETB    RST
        DELAY5uS
        CLR     RST
        RET
```

;**等待** IC **卡操作完毕**

```
WRMOPTION:
        CLR     CLK
        DELAY5uS
        SETB    IO
        NOP
        NOP
        MOV     C,IO
        JC      WRMOK
        DELAY5uS
        SETB    CLK
        DELAY5uS
        SJMP    WRMOPTION
WRMOK:  RET
```

;**发送命令子程序**

;**入口参数：**R4**一命令字** R5**一地址数** R6**一数据字节**

```
SENDCOM:
        SETB    IO              产生开始信号
        DELAY5uS
        SETB    CLK
        DELAY5uS
        CLR     IO
        DELAY5uS
        CLR     CLK
        MOV     A,R4            发送命令
```

```
                ACALL     SENDBYTE
                MOV       A,R5
                ACALL     SENDBYTE
                MOV       A,R6
                ACALL     SENDBYTE
                CLR       IO                    产生结束信号
                DELAY5uS
                SETB      CLK
                DELAY5uS
                SETB      IO
                DELAY5uS
                RET

;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
;软件包接口命令

;读主存储器子程序
;入口参数为 STARTADR，BYTENUM 读取字节数
;出口数据依次保存在 RDBUF 处,卡号在 CARDBAK
IRDMM_4442:
                ACALL     ANRST
                MOV       R4,#30H               装入命令字
                MOV       R5,STARTADR
                ACALL     SENDCOM               发送命令
                MOV       R2,BYTENUM
                MOV       R0,#RDBUF
RDMM_L1:        ACALL     RCVBYTE
                MOV       @R0,A
                INC       R0
                DJNZ      R2,RDMM_L1
                ACALL     BREAK
                RET

;写主存储器子程序
;入口参数为 STARTADR，BYTENUM 读取字节数
;要写入的数据依次保存在 WRBUF 处
IWRMM_4442:
                ACALL     ANRST
                MOV       R5,STARTADR
                MOV       R0,#WRBUF
                MOV       R2,BYTENUM
WRMM_L1:        MOV       R4,#38H               命令字
                MOV       A,@R0
                MOV       R6,A
```

```
                ACALL     SENDCOM
                INC       R5
                INC       R0
                ACALL     WRMOPTION
                DJNZ      R2,WRMM_L1
                RET

;读加密存储器
;读出数据放在 RDBUF（4 个字节）
IRDSCM_4442:
                ACALL     ANRST
                MOV       R4,#31H
                ACALL     SENDCOM       发送读加密存储器指令
                MOV       R2,#04H
                MOV       R0,#RDBUF
RDSCM_L1:       ACALL     RCVBYTE       读出加密字节
                MOV       @R0,A
                INC       R0
                DJNZ      R2,RDSCM_L1
                CLR       CLK
                DELAY5uS
                RET

;写加密存储器
;将 WRBUF 缓冲区的数据
IWRSCM_4442:
                ACALL     ANRST
                MOV       R5,STARTADR   要写入的地址
                MOV       R0,#WRBUF     要写入数据区指针
                MOV       R2,BYTENUM    写入字节数
WRSCM_L1:       MOV       R4,#39H
                MOV       06H,@R0
                ACALL     SENDCOM
                INC       R5
                INC       R0
                ACALL     WRMOPTION
                DJNZ      R2,WRSCM_L1
                RET

;读保护存储器
;将保护存储器 32 位数据读入 RDBUF
IRDPRM_4442:
                ACALL     ANRST
                MOV       R4,#34H
```

```
                ACALL       SENDCOM
                MOV         R0,#RDBUF
                MOV         R2,BYTENUM
                MOV         R5,STARTADR
RDPRM_L1:       ACALL       RCVBYTE          接收保护存储器数据字节
                MOV         @R0,A
                INC         R0
                DJNZ        R2,RDPRM_L1
                ACALL       BREAK
                RET

;写保护存储器
;将 WRBUF 的 X 位数据写入保护存储器，对保护数据区进行保护
IWRPRM_4442:
                ACALL       ANRST
                MOV         R5,STARTADR
                MOV         R0,#WRBUF
                MOV         R2,BYTENUM
WRPRM_L1:       MOV         R4,#3CH
                MOV         06H,@R0
                ACALL       SENDCOM
                INC         R5
                INC         R0
                ACALL       WRMOPTION
                DJNZ        R2,WRPRM_L1
                RET

; 比较校检密码数据
; 密码字节依次写入 WRBUF（3 字节）
ICHKPSW_4442:
                MOV         A,WRBUF
                MOV         WRBUF+4,A
                ACALL       IRDSCM_4442
                MOV         A,RDBUF
                ANL         A,#07H
                JZ          ISBADCARD        ;卡已报废
                RRC         A
                JNC         CHKPSW1
                MOV         A,RDBUF          最低位为 1
                ANL         A,#06H           则将其置为 0，方可校对密码
                JZ          HAVEONECHANCE    若只有一次机会，退出操作
                SJMP        CHKPSW2
CHKPSW1:        RRC         A                判断第 2 位是否为 1
                JNC         HAVEONECHANCE    ;不为 1 则此卡中有一次机会，退出操作
```

```
                MOV             A,RDBUF
                ANL             A,#05H                  将第二位置为 0
                JZ              HAVEONECHANCE
CHKPSW2:        MOV             STARTADR,#00H           进行校验操作
                MOV             WRBUF,A
                MOV             BYTENUM,#01H
                ACALL           IWRSCM_4442             回写 EC
                MOV             A,WRBUF+4
                MOV             WRBUF,A
                MOV             R2,#03H
                MOV             R5,#01H
                MOV             R0,#WRBUF
CHKPSWL1:       MOV             A,@R0
                MOV             R4,#33H
                MOV             R6,A
                ACALL           SENDCOM
                ACALL           WRMOPTION
                INC             R0
                INC             R5
                DJNZ            R2,CHKPSWL1             三字节密码校验完毕
                MOV             WRBUF,#0FFH
                MOV             BYTENUM,#01H
                MOV             STARTADR,#00H
                ACALL           IWRSCM_4442             擦除错误计数器
                ACALL           IRDSCM_4442
                MOV             A,RDBUF
                ANL             A,#07H
                CJNE            A,#07H,IS_FAIL
                MOV             A,#00H
                RET
IS_FAIL:        MOV             A,#02H
                RET
ISBADCARD:      MOV             A,#03H
                RET
HAVEONECHANCE:      MOV         A,#04H
                    RET

;占用了 ACC，R0，R2，R3，R4，R5，R6 及 CY 位。
;硬件占 3 个 I/O 口,RST，CLK，IO，IRQ。
```

3、C51 软件包

```
/*********************************************************************
                                IC4442_C51.C
   此程序是 IC 卡 SLE4442 的软件包，把对卡的操作如读主存储器，写主存储器，
校验密码，读加密存储器等集成成三个函数，使用了枚举变量的方法把操作存
储器区分开。SLE4442 为 2 线连接协议，串行接口满足 ISO7816 同步传送协议，即
其时序和 I2C 基本一致。
        注意:函数是采用软件延时的方法产生 CLK 脉冲,固对高晶振频率要作
一定的修改....(本例是 1us 机器周期,即晶振频率要小于 12MHZ)
*********************************************************************/
#include  <reg764.h>                  /*头文件的包含*/
#include <intrins.h>

#define   uchar unsigned char  /*宏定义*/
#define    MAM          0                  /*定义主存储器代号*/
#define    SCM          1                  /*定义加密存储器代号*/
#define    PRM          2                  /*定义保护存储器代号*/
#define    _Nop()    _nop_()          /*定义空指令*/
```

```
#define    DELAY5us()  _Nop();_Nop();_Nop();_Nop();_Nop()

/*端口位定义*/
sbit RST=P1^0;
sbit  IO=P1^6;
sbit CLK=P1^1;


/*****************************************************************
                        起动总线函数
函数原型: void    Start_COM();
功能:       启动发送命令起始条件.
*****************************************************************/
void Start_COM()
{
    IO=1;                   /*发送起始条件的数据信号*/
    _Nop();
    CLK=1;
    DELAY5us();         /*起始条件建立时间大于 4.7us,延时*/
    IO=0;                   /*发送起始信号*/
    DELAY5us();         /*起始条件锁定时间大于 4μs*/
    CLK=0;                  /*钳住总线，准备发送或接收数据  */
    _Nop();
    _Nop();
}


/*****************************************************************
                        结束总线函数
函数原型: void    Stop_COM();
功能:       命令发送结束信号。
*****************************************************************/
void Stop_COM()
{
    IO=0;           /*发送结束条件的数据信号*/
    _Nop();       /*发送结束条件的时钟信号*/
    CLK=1;          /*结束条件建立时间大于 4μs*/
    DELAY5us();
    IO=1;           /*发送总线结束信号*/
    _Nop();
    _Nop();
}
```

```
/*********************************************************************
                        字节数据传送函数
函数原型: void    SendByte(uchar  c);
功能:    将数据 c 发送出去,可以是命令,也可以是数据.
*********************************************************************/
void    SendByte(uchar  c)
{
 uchar  BitCnt;

 for(BitCnt=0;BitCnt<8;BitCnt++)  /*要传送的数据长度为 8 位*/
       {
         if((c>>BitCnt)&0x01)IO=1;       /*判断发送位*/
                else     IO=0;
         _Nop();_Nop();
         CLK=1;                                  /*置时钟线为高，通知被控器开始接收数据位*/
         DELAY5us();                            /*保证时钟高电平周期大于 4μs*/
        CLK=0;
       }
}


/*********************************************************************
                        字节数据接收函数
函数原型: uchar    RcvByte();
功能:    用来接收从卡传来的数据.
*********************************************************************/
uchar    RcvByte()
{
   uchar  retc;
   uchar  BitCnt;

   retc=0;
    for(BitCnt=0;BitCnt<8;BitCnt++)
          {
             CLK=0;                                 /*置时钟线为低，准备接收数据位*/
             IO=1;                                 /*置数据线为输入方式*/
             DELAY5us();                          /*时钟低电平周期大于 4.7μs*/
             CLK=1;                                 /*置时钟线为高使数据线上数据有效*/
             _Nop();
             _Nop();
             retc=retc>>1;
            if(IO==1)retc=retc+0x80;  /*读数据位,接收的数据位放入 retc 中  */
```

```
            _Nop();  _Nop();
          }
    CLK=0;
    _Nop();_Nop();
    return(retc);
}

/*****************************************************************
                        复位和复位响应函数
函数原型：    void    AnRst();
功能：          复位 IC 卡并接收响应字节
*****************************************************************/
void AnRst()
{
    RST=1;              /*产生复位时序*/
    DELAY5us();
    CLK=1;
    DELAY5us();
    CLK=0;
    DELAY5us();
    RST=0;
    _Nop();

    RcvByte();      /*读出 32 字节响应数据*/
   RcvByte();
    RcvByte();
    RcvByte();
}



/*****************************************************************
                        发送 4442 处理脉冲函数
函数原型：    void    WrmOption();
功能：          发送处理模式指令后要调用此程序发送脉冲.
*****************************************************************/
void    WrmOption()
{
  while(1)
     {
       CLK=0;
       DELAY5us();
       IO=1;
       _Nop();_Nop();
```

```
        if(IO==1)break;         /*没有处理完则继续发送脉冲*/
        CLK=1;
        DELAY5us();
      }
}

/*********************************************************************
                          中止操作函数
函数原型：    void    BreakN();
功能：          中止当前操作。
*********************************************************************/
void    BreakN()
{
        CLK=0;
        DELAY5us();
        RST=1;                          /*发出中止操作的时序*/
        DELAY5us();
        RST=0;
}

/*********************************************************************
                          命令发送函数
函数原型：  void        SendCOM(ucahr  com1,ucahr com2,uchar com3);
功能：          负责起动命令，发送 3 字节命令字
                结束命令。
*********************************************************************/
void   SendCOM(ucahr com1,uchar com2,uchar com3)
{
    Start_COM();
   SendByte(com1);  /*连续发送 3 字节指令*/
    SendByte(com2);
    SendByte(com3);
    Stop_COM();
}

/*********************************************************************
                          SLE4442 卡读数据函数
函数原型: bit IRcvdat_4442(uchar area,ucahr addr,uchar num,uchar buf[]);
功能:           对 SLE4442 卡进行读操作，area 为存储器类型，addr 为起始地址，
                num 为读取数据字节数，buf[]为数据缓冲区指针。
说明：          操作成功返回 1，参数 area 错误返回 0。使用前用判断卡插好没有?
```

```
*********************************************************************/
bit IRcvdat_4442(uchar area,uchar addr,uchar num,uchar buf[])
{
 uchar i;
  switch(area)
        {
           case  MAM:    AnRst();                /*复位 SLM4442 卡，接收复位响应*/
                         SendCOM(0X30,addr,0x00);    /*读主存储器*/
                         for(i=0;i<num;i++)
                            {
                              *buf=RcvByte();
                              buf++;
                            }
                         BreakN();
                         break;
           case  SCM:    AnRst();
                         SendCOM(0x31,0x00,0x00)
                         for(i=0;i<num;i++)
                            {
                              *buf=RcvByte();
                                buf++;
                            }
                         BreakN();
                         break;
           case  PRM:    AnRst();
                         SendCOM(0x34,0x00,0x00)
                        for(i=0;i<num;i++)
                              {
                                 *buf=RcvByte();
                                 buf++;
                              }
                        BreakN();
                        break;
          default:    return(0);
        }
return(1);
}

/*********************************************************************
                              SLE4442 卡写数据函数
函数原型: bit ISenddat_4442(uchar area,ucahr addr,uchar num,uchar buf[]);
功能:        对 SLE4442 卡进行写操作，area 为存储器类型，addr 为起始地址，
             num 为读取数据字节数，buf[]为数据缓冲区指针。
说明:        操作成功返回 1，参数 area 错误返回 0。使用前用判断卡插好没有？
```

```
*********************************************************************/
bit ISenddat_4442(uchar area,uchar addr,uchar num,uchar buf[])
{
 uchar i;
  switch(area)
        {
           case  MAM:        AnRst();
                              for(i=0;i<num;i++)
                              {
                                SendCOM(0X38,addr+i,*buf);    /*写主存储器*/
                                buf++;
                                WrmOption();                          /*发送操作脉冲*/
                              }
                            break;
           case  SCM:      AnRst();
                             for(i=0;i<num;i++)
                              {
                                 SendCOM(0x39,addr+i,*buf)
                                 buf++;
                                 WrmOption();
                              }
                            break;
           case  PRM:    AnRst();
                           for(i=0;i<num;i++)
                               {
                                  SendCOM(0x3c,addr+i,*buf);
                                  buf++;
                                  WrmOption();
                               }
                            break;
          default:    return(0);
        }
return(1);
}

/*********************************************************************
                              SLE4442 卡校验密码函数
函数原型: uchar IChkpsw_4442(uchar psw1,uchar psw2,uchar psw3);
功能:        进行 SLE4442 卡进行密码核对，核对后方能进行写操作。
说明:        操作成功返回 0x00，卡无效或卡损坏返回 0x01,密码错误返
               回 0x02，卡只剩 1 次机会返回 0x03.
*********************************************************************/
uchar   IChkpsw_4442(uchar psw1,ucahr psw2,ucahr psw3)
{
```

```
  uchar  ec[1];
  IRcvdat_4442(SCM,0X00,1,ec);
  if((ec[0]&0x07)==0)return(0x01);     /*卡损坏*/
  if((ec[0]&0x06)==0)
          {
            if((ec[0]&0x05)==0)return(0x03);
               else  ec[0]=0x05;                /*EC 的 D0 位为 0 时，判 D1 位为 1? */
          }
     else  ec[0]=0x06;                          /*EC 的 D0 位为 1*/
  AnRst();
  SendCOM(0x39,0x00,ec[0]);                  /*回写 EC 字节*/
  WrmOption();
  SendCOM(0x33,0x01,psw1);
 WrmOption();
  SendCOM(0x33,0x02,psw2);
  WrmOption();
  SendCOM(0X33,0X03,psw3);
  WrmOption();
  SendCOM(0X39,0X00,0Xff);                   /*修改 EC 值*/
  WrmOption();
  IRcvdat_4442(SCM,0X00,1,ec[0]);
  if((ec[0]&0x07)!=0x07)return(0x02);
  return(0x00);
}

                /*      END             */
```